

Fourth-order two-stage explicit exponential integrators for solving differential equations

Vu Thai Luan

*School of Natural Sciences, University of California, Merced, 5200 North Lake Road,
Merced, CA 95343, USA*

Abstract

Among the family of fourth-order time integration schemes, the two-stage Gauss–Legendre method, which is an implicit Runge–Kutta method based on collocation, is the only superconvergent. The computational cost of this implicit scheme for large systems, however, is very high since it requires solving a nonlinear system at every step. Surprisingly, in this work we show that one can construct and prove convergence results for exponential methods of order four which use two stages only. Specifically, we derive two new fourth-order two-stage exponential Rosenbrock schemes for solving large systems of differential equations. Moreover, since the newly schemes are not only superconvergent but also fully explicit, they clearly offer great advantages over the two-stage Gauss–Legendre method as well as other time integration schemes. Numerical experiments are given to demonstrate the efficiency of the new integrators.

Keywords: Exponential integrators, exponential Rosenbrock methods, nonstiff problems, stiff problems, superconvergence

1. Introduction

Finding numerical solutions to time-dependent PDEs usually requires the time discretization of large systems of differential equations, which can be cast in the form

$$u'(t) = F(u(t)), \quad u(t_0) = u_0. \quad (1.1)$$

Along with the development of numerical analysis, many methods have been designed for solving (1.1) numerically. Depending on the characteristics of

Email address: `vluan@ucmerced.edu` (Vu Thai Luan)

each problem, one has to choose the right method. Nonstiff problems are usually integrated by using classical techniques such as explicit Runge–Kutta methods, multistep methods, and general linear methods (see [1]). The situation for stiff problems, however, is much more involved due to the fact that the Jacobian matrix often has a large norm or is even an unbounded operator. In this case, explicit methods have to face with stability issues. They usually lack stability and are required to use extremely small time steps. To avoid this issue, various kinds of implicit methods have been proposed such as implicit RungeKutta methods (e.g., Gauss, Radau IA and IIA), BDF methods, Rosenbrock-type methods, just to name a few. For details of such methods we refer the reader to the excellent monograph [2]. The downside of implicit methods, however, is their computational costs that are usually too high caused by solving large nonlinear system at every step. In order to overcome the two mentioned issues of such classical explicit and implicit methods, exponential integrators has been introduced (see the review paper [3] for details). This field has grown significantly since 1998 and it has been shown that the integrators are highly competitive, see for example [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. High-order exponential integrators for stiff problems have been proposed in [15].

In this work, we pay attention to a very complete and good candidate for solving stiff problems, the so-called exponential Rosenbrock methods, which is first proposed in [16]. The idea is, first to make a continuous linearization of the vector field F along the numerical solution u_n of (1.1) (due to Pope [17]) leading to semilinear problems

$$u'(t) = J_n u(t) + g_n(u(t)) \quad (1.2)$$

with the Jacobian J_n and the nonlinearity g_n are

$$J_n = \frac{\partial F}{\partial u}(u_n), \quad g_n(u) = F(u) - J_n u, \quad (1.3)$$

and then to apply exponential Runge–Kutta methods [6] to (1.2) which resulted in exponential Rosenbrock methods. They have been studied intensively in a series of papers [16, 18, 9, 13]. Methods up to order 6 have been derived in [13] and the stiff order conditions for methods up to arbitrary order are given in [19]. One of the great advantages of exponential Rosenbrock methods is that the Jacobian of the nonlinearities $g_n(u)$ vanishes at the numerical solution u_n (see (2.10)). This improves the stability, simplifies the

stiff order conditions, and thus allows one to construct high-order methods with a few stages only. For instance, we mention the class of 1-stage method of order 2 (considered as a superconvergence scheme), namely the exponential Rosenbrock-Euler method (see [18]) and the class of 3-stage method of order 5, `exprb53s3` (see [9]).

Our aim in this paper is to construct, analyze, and implement a class of fourth-order 2-stage explicit exponential Rosenbrock methods. This is motivated by the fact a 3-stage method can get the maximum order $p = 5$, see [9]. Moreover, it is not clear on the existence of a 2-stage method which has superconvergence property (order 4) both in the context of nonstiff and stiff problems. By taking further investigate order conditions both in the classical and stiff sense, we will show that one can construct such methods.

The remainder of the paper is organized as follows. In Section 2, we recall the exponential Rosenbrock schemes (including 2-stage methods) and present our motivation to this work. The construction of classical fourth-order 2-stage schemes is given in Section 3, where we give the classical order conditions (Lemma 3.1), show the convergence result (Theorem 3.1), and derive the scheme `exprb42N` (see (3.16)). Inspired by these results, in Section 4 we show that, under the regularity assumptions on the problem, it is even possible to construct a stiffly accurate fourth-order 2-stage scheme. The main results of this section are Lemma 4.1 (stiff order conditions), Theorem 4.1 (convergence), and the scheme `exprb42` (see (4.12)). Section 5 discusses variable stepsizes implementation for the two newly constructed integrators. Finally, in Section 6 we verify the convergence results and show the efficiency of the two new integrators on a set of numerical examples.

2. Numerical method and motivation

We start off by recalling the explicit exponential Rosenbrock-type methods for solving (1.1), see [18]:

$$U_{ni} = u_n + c_i h \varphi_1(c_i h J_n) F(u_n) + h \sum_{j=2}^{i-1} a_{ij}(h J_n) D_{nj}, \quad (2.1a)$$

$$u_{n+1} = u_n + h \varphi_1(h J_n) F(u_n) + h \sum_{i=2}^s b_i(h J_n) D_{ni} \quad (2.1b)$$

with $D_{ni} = g_n(U_{ni}) - g_n(u_n)$, $2 \leq i \leq s$. Here $u_n \approx u(t_n)$, c_i are the nodes, s is the number of stages, $U_{ni} \approx u(t_n + c_i h)$, J_n and g_n are given in (1.3),

$h = t_{n+1} - t_n > 0$ denotes the time step. The coefficients $a_{ij}(z)$ and $b_i(z)$ are usually chosen as linear combinations of the corresponding entire functions $\varphi_k(c_i z)$ and $\varphi_k(z)$, where

$$\varphi_0(z) = e^z, \quad \varphi_k(z) = \int_0^1 e^{(1-\theta)z} \frac{\theta^{k-1}}{(k-1)!} d\theta, \quad k \geq 1. \quad (2.2)$$

These functions satisfy the relation

$$\varphi_k(z) = \frac{1}{k!} + z\varphi_{k+1}(z), \quad k \geq 0. \quad (2.3)$$

So far, it is known that the one-stage ($s = 1$) second-order method, the so-called exponential Rosenbrock-Euler method

$$u_{n+1} = u_n + h\varphi_1(hJ_n)F(u_n), \quad (2.4)$$

is the only superconvergent exponential integrator. It is shown in [18] that the 2-stage schemes, which read as

$$U_{n2} = u_n + c_2 h \varphi_1(c_2 h J_n) F(u_n), \quad (2.5a)$$

$$u_{n+1} = u_n + h\varphi_1(hJ_n)F(u_n) + hb_2(hJ_n)(g_n(U_{n2}) - g_n(u_n)), \quad (2.5b)$$

can attain third-order accuracy, see for example the scheme **exprb32** given in [18]. Obviously, from this one can easily derive a corresponding classical third-order scheme. However, the question whether or not a 2-stage method of order 4 exists is still open. On the other hand, in [9] it is shown that a 3-stage method can reach the maximum order $p = 5$. Therefore, our aim in this work is to answer the question of superconvergence for the class of 2-stage methods. It is thus important to further investigate order conditions for such a 2-stage scheme (2.5), both in the classical and stiffly accurate situation. For this purpose, our idea is to analyze local errors directly as done in [9]. Namely, we will study one step integration scheme (2.5) with the initial values on the exact solution $\tilde{u}_n = u(t_n)$, i.e.

$$\bar{U}_{n2} = \tilde{u}_n + c_2 h \varphi_1(c_2 h \tilde{J}_n) F(\tilde{u}_n), \quad (2.6a)$$

$$\bar{u}_{n+1} = \tilde{u}_n + h\varphi_1(h\tilde{J}_n)F(\tilde{u}_n) + hb_2(h\tilde{J}_n)(\tilde{g}_n(\bar{U}_{n2}) - \tilde{g}_n(\tilde{u}_n)). \quad (2.6b)$$

Similarly to (1.3), here

$$\tilde{J}_n = \frac{\partial F}{\partial u}(\tilde{u}_n), \quad \tilde{g}_n(u) = F(u) - \tilde{J}_n u, \quad (2.7)$$

which are resulted from the linearization of (1.1) at \tilde{u}_n , i.e.

$$u'(t) = \tilde{J}_n u(t) + \tilde{g}_n(u(t)). \quad (2.8)$$

Let

$$\tilde{e}_{n+1} = \bar{u}_{n+1} - \tilde{u}_{n+1} \quad (2.9)$$

denote the local error, i.e., the error of the numerical solution after one step with initial value on the exact solution \tilde{u}_n . Since the structure of (2.6) allows to treat the linear part of (1.2) exactly (see Remark 1 below) and the fact that

$$\frac{\partial \tilde{g}_n}{\partial u}(\tilde{u}_n) = \frac{\partial}{\partial u}(F(u) - \tilde{J}_n u)(\tilde{u}_n) = \tilde{J}_n - \tilde{J}_n = 0, \quad (2.10)$$

it is hoped that one can further simplify order conditions and derive from that the right coefficient $b_2(h\tilde{J}_n)$, which gives order of consistency 5 for the local error, i.e. $\tilde{e}_{n+1} = \mathcal{O}(h^5)$.

For the remaining of the paper, we will focus on both cases: nonstiff and stiff problems. Our analysis will be performed in a Banach space X with norm $\|\cdot\|$.

3. Construction of classical fourth-order 2-stage exponential Rosenbrock schemes

In this section we consider the case where the vector field $F(u)$ is a nonlinear function with a moderate Lipschitz constant. In other words, the problem (1.1) is supposed to be nonstiff. We thus can make use of the following assumption.

Assumption 1. Suppose that (1.1) possesses a sufficiently smooth solution $u : [0, T] \rightarrow X$, with derivatives in X and that $F : X \rightarrow X$ is sufficiently often Fréchet differentiable in a strip along the exact solution. All occurring derivatives are assumed to be bounded.

Clearly, under this assumption, $g_n(u) = F(u) - J_n u$ is also sufficiently often Fréchet differentiable as well as satisfies the Lipschitz condition in a strip along the exact solution.

We note for later use that under Assumption 1 one can expand $\varphi_1(c_2 h \tilde{J}_n)$

(by using the recurrence relation (2.3)) and $b_2(h\tilde{J}_n)$ appearing in (2.6) as

$$\varphi_1(c_2 h \tilde{J}_n) = \sum_{k \geq 0} \frac{(c_2 h \tilde{J}_n)^k}{(k+1)!} = I + \frac{1}{2!} c_2 h \tilde{J}_n + \frac{1}{3!} c_2^2 h^2 \tilde{J}_n^2 + \mathcal{O}(h^3), \quad (3.1a)$$

$$b_2(h\tilde{J}_n) = \sum_{k \geq 0} \beta_k (h\tilde{J}_n)^k = \beta_0 I + \beta_1 h \tilde{J}_n + \mathcal{O}(h^2). \quad (3.1b)$$

We now derive an expansion of the numerical solution \bar{u}_{n+1} .

3.1. Expansion of the numerical solution

Let $\tilde{u}'_n, \tilde{u}''_n, \tilde{u}'''_n$ denote the first, second, and third derivative of the exact solution $u(t)$ of (1.1) evaluated at time t_n , respectively. We further denote $\frac{\partial \tilde{g}_n}{\partial u}(u)$, $\frac{\partial^2 \tilde{g}_n}{\partial u^2}(u)$, and $\frac{\partial^3 \tilde{g}_n}{\partial u^3}(u)$ by $\tilde{g}'_n(u)$, $\tilde{g}''_n(u)$, and $\tilde{g}_n^{(3)}(u)$, respectively.

By using (2.10) and differentiating the equation (2.8) twice, we obtain

$$\tilde{J}_n \tilde{u}'_n = \tilde{u}''_n, \quad \tilde{J}_n^2 \tilde{u}'_n = \tilde{u}'''_n - \tilde{g}''_n(\tilde{u}_n)(\tilde{u}'_n, \tilde{u}'_n). \quad (3.2)$$

Inserting $F(\tilde{u}_n) = \tilde{u}'_n$ and (3.1a) into (2.6a) with the help of the identities in (3.2) gives

$$\bar{U}_{n2} = \tilde{u}_n + c_2 h \tilde{u}'_n + \frac{1}{2!} c_2^2 h^2 \tilde{u}''_n + \frac{1}{3!} c_2^3 h^3 (\tilde{u}'''_n - \tilde{g}''_n(\tilde{u}_n)(\tilde{u}'_n, \tilde{u}'_n)) + \mathcal{O}(h^4). \quad (3.3)$$

By employing (3.3) and (2.10), we next expand $\tilde{g}_n(\bar{U}_{n2})$ in a Taylor series at \tilde{u}_n to get

$$\begin{aligned} \tilde{g}_n(\bar{U}_{n2}) - \tilde{g}_n(\tilde{u}_n) &= \frac{1}{2!} c_2^2 h^2 \tilde{g}''_n(\tilde{u}_n)(\tilde{u}'_n, \tilde{u}'_n) + \frac{1}{3!} c_2^3 h^3 (\tilde{g}_n^{(3)}(\tilde{u}_n)(\tilde{u}'_n, \tilde{u}'_n, \tilde{u}'_n) \\ &\quad + 3\tilde{g}''_n(\tilde{u}_n)(\tilde{u}'_n, \tilde{u}''_n)) + \mathcal{O}(h^4). \end{aligned} \quad (3.4)$$

Inserting (3.1b) and (3.4) into (2.6b) yields the following expansion

$$\begin{aligned} \bar{u}_{n+1} &= \tilde{u}_n + h \varphi_1(h\tilde{J}_n) F(\tilde{u}_n) + h^3 \frac{1}{2!} \beta_0 c_2^2 \tilde{g}''_n(\tilde{u}_n)(\tilde{u}'_n, \tilde{u}'_n) \\ &\quad + h^4 \left(\frac{1}{3!} \beta_0 c_2^3 (\tilde{g}_n^{(3)}(\tilde{u}_n)(\tilde{u}'_n, \tilde{u}'_n, \tilde{u}'_n) + 3\tilde{g}''_n(\tilde{u}_n)(\tilde{u}'_n, \tilde{u}''_n)) + \frac{1}{2!} \beta_1 c_2^2 \tilde{J}_n \tilde{g}''_n(\tilde{u}_n)(\tilde{u}'_n, \tilde{u}'_n) \right) \\ &\quad + \mathcal{O}(h^5). \end{aligned} \quad (3.5)$$

Remark 1. It can be seen from the expansion of the numerical solution in (3.5) that we do not expand the term $\varphi_1(h\tilde{J}_n)$ in a power series of $h\tilde{J}_n$ as done for $\varphi_1(c_2h\tilde{J}_n)$ in (3.1a). The reason for keeping that term is because the sum of the first two terms in (3.5) can be rewritten, by using the fact that $\varphi_1(z) = (e^z - 1)/z$ and $F(\tilde{u}_n) = \tilde{J}_n\tilde{u}_n + \tilde{g}_n(\tilde{u}_n)$, as

$$\tilde{u}_n + h\varphi_1(h\tilde{J}_n)F(\tilde{u}_n) = e^{h\tilde{J}_n}\tilde{u}_n + h\varphi_1(h\tilde{J}_n)\tilde{g}_n(\tilde{u}_n) \quad (3.6)$$

which can be used to treat the linear part of (1.2) exactly as seen in the expansion of the exact solution as follows.

3.2. Expansion of the exact solution

Expressing the exact solution of (1.2) at time t_{n+1} by the variation-of-constants formula gives

$$\tilde{u}_{n+1} = u(t_{n+1}) = e^{h\tilde{J}_n}\tilde{u}_n + h \int_0^1 e^{(1-\theta)h\tilde{J}_n} \tilde{g}_n(u(t_n + \theta h)) d\theta \quad (3.7)$$

which can be rewritten as

$$\tilde{u}_{n+1} = e^{h\tilde{J}_n}\tilde{u}_n + h \int_0^1 e^{(1-\theta)h\tilde{J}_n} \tilde{g}_n(\tilde{u}_n) d\theta + h \int_0^1 e^{(1-\theta)h\tilde{J}_n} (\tilde{g}_n(u(t_n + \theta h)) - \tilde{g}_n(\tilde{u}_n)) d\theta. \quad (3.8)$$

One can realize that the sum of the first two terms of (3.8) is exactly equal to (3.6). Next, by employing (2.10) we expand $\tilde{g}_n(u(t_n + \theta h))$ in a Taylor series at \tilde{u}_n and insert the obtained results into the third term of (3.8) as done in [9, Sec. 3.2], which finally gives

$$\begin{aligned} \tilde{u}_{n+1} = & \tilde{u}_n + h\varphi_1(h\tilde{J}_n)F(\tilde{u}_n) + h^3\varphi_3(h\tilde{J}_n)\tilde{g}_n''(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n') \\ & + h^4\varphi_4(h\tilde{J}_n)(\tilde{g}_n^{(3)}(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n', \tilde{u}_n') + 3\tilde{g}_n''(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n'')) + \mathcal{O}(h^5). \end{aligned} \quad (3.9)$$

We now insert the following expansions

$$\begin{aligned} \varphi_3(h\tilde{J}_n) &= \frac{1}{3!}I + \frac{1}{4!}h\tilde{J}_n + (h\tilde{J}_n)^2\varphi_5(h\tilde{J}_n) \\ \varphi_4(h\tilde{J}_n) &= \frac{1}{4!}I + (h\tilde{J}_n)\varphi_5(h\tilde{J}_n) \end{aligned} \quad (3.10)$$

obtained by using (2.3) into (3.9) to get

$$\begin{aligned}\tilde{u}_{n+1} = \tilde{u}_n &+ h\varphi_1(h\tilde{J}_n)F(\tilde{u}_n) + h^3 \frac{1}{3!} \tilde{g}_n''(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n') \\ &+ h^4 \frac{1}{4!} \left(\tilde{g}_n^{(3)}(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n', \tilde{u}_n') + 3\tilde{g}_n''(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n'') + \tilde{J}_n \tilde{g}_n''(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n') \right) + \mathcal{O}(h^5).\end{aligned}\tag{3.11}$$

With this expansion of the exact solution at hand, we are now ready to derive (classical) order conditions for 2-stage methods of order 4.

3.3. Local error and order conditions for fourth-order 2-stage methods

By subtracting (3.11) from (3.5), it is straightforward to derive the following result for the local error \tilde{e}_{n+1} .

Lemma 3.1. *Under Assumption 1, an explicit 2-stage exponential Rosenbrock scheme (2.5) in which the coefficient $b_2(hJ_n)$ can be expanded as (3.1b), has order of consistency five, i.e. the local error $\tilde{e}_{n+1} = \mathcal{O}(h^5)$ if the following order conditions are fulfilled*

$$\frac{\beta_0 c_2^2}{2!} = \frac{1}{3!}, \quad \frac{\beta_0 c_2^3}{3!} = \frac{1}{4!}, \quad \frac{\beta_1 c_2^2}{2!} = \frac{1}{4!}, \tag{3.12}$$

that is equivalent to

$$c_2 = \frac{3}{4}, \quad \beta_0 = \frac{16}{27}, \quad \beta_1 = \frac{4}{27}. \tag{3.13}$$

Here the remainder term of \tilde{e}_{n+1} , which is hidden behind the Landau notation $\mathcal{O}(\cdot)$, is bounded by Ch^5 with a constant C that depends on $\|\tilde{J}_n\|$. \square

The result of Lemma 3.1 implies that the coefficient $b_2(hJ_n)$ of (2.5) must satisfy the following expansion

$$b_2(hJ_n) = \frac{16}{27}I + \frac{4}{27}hJ_n + \mathcal{O}(h^2). \tag{3.14}$$

in order for the method to have order of consistency five.

3.4. Convergence result

In the following we show that such a scheme (2.5) that takes $c_2 = 3/4$ and fulfills (3.14) is indeed convergent with a global error of order 4.

Theorem 3.1. *Let the initial value problem (1.1) satisfies Assumption 1. Consider for its numerical solution an explicit 2-stage exponential Rosenbrock scheme (2.5) with $b_2(hJ_n)$ satisfies (3.14) and the node $c_2 = \frac{3}{4}$ (fulfilling the order conditions in (3.13)). Then, the method converges with order four, i.e.*

$$\|u_n - u(t_n)\| \leq Ch^4 \quad (3.15)$$

on $t_0 \leq t_n = t_0 + nh \leq T$ with a constant C that depends on n and h .

Proof. It is remaining to show that the numerical scheme (2.5) is stable. This is straightforward due to the fact that, under Assumption 1, the Jacobian $J(u) = \frac{\partial F(u)}{\partial u}$ also satisfies the Lipschitz condition in a strip along the exact solution u . Another possibility is to employ the stability condition of exponential Rosenbrock methods which is recalled in (4.13) in Section 4.3 below. We thus omit the details. \square

3.5. Derivation of classical fourth-order 2-stage schemes

Clearly, a 2-stage scheme (2.5) is derived if the coefficient $b_2(hJ_n)$ is identified. Since $b_2(hJ_n)$ is usually chosen as linear combinations of some matrix functions $\varphi_k(hJ_n)$, the condition (3.14) determines explicitly such a linear combination. For example, one can choose $b_2(hJ_n)$ as a linear combination of $\varphi_1(hJ_n)$ and $\varphi_2(hJ_n)$ as $b_2(hJ_n) = -\frac{8}{27}\varphi_1(hJ_n) + \frac{48}{27}\varphi_2(hJ_n)$ resulting in the following scheme which will be called **exprb42N**:

$$U_{n2} = u_n + \frac{3}{4}h\varphi_1\left(\frac{3}{4}hJ_n\right)F(u_n), \quad (3.16a)$$

$$u_{n+1} = u_n + h\varphi_1(hJ_n)F(u_n) + h\left(-\frac{8}{27}\varphi_1(hJ_n) + \frac{48}{27}\varphi_2(hJ_n)\right)(g_n(U_{n2}) - g_n(u_n)). \quad (3.16b)$$

Note that one can derive many other 2-stage fourth-order schemes like (3.16) as long as $b_2(hJ_n)$ satisfies condition (3.14).

4. Construction of a stiff fourth-order 2-stage exponential Rosenbrock scheme

It should be mentioned that our convergence analysis presented in Section 3 cannot be applied if the Jacobian J_n has a large norm or is even unbounded operator. The reason for that is simply because Assumption 1 and thus the expansions (3.1) and (3.10) are no longer valid. Unfortunately,

this is usually the situation of stiff problems arising when discretizing the space dimension of many time dependent PDEs. Examples of such problems are diffusion-reaction equations, the heat equations, just to mention a few. Therefore, in this section our aim is to design a 2-stage exponential Rosenbrock scheme of the form (2.5) that is superconvergent and works for such stiff problems. We will focus on the common case where the vector field $F(u)$ can be decomposed into two parts: the linear part which is stiff and the nonlinear part which is nonstiff, namely

$$u'(t) = F(u(t)) = Au(t) + g(u(t)), \quad u(t_0) = u_0. \quad (4.1)$$

In the subsequent analysis, we will use the framework of strongly continuous semigroups in the Banach space X (for instance, see [20, 21]) to handle this type of stiff problems. In particular, throughout this section the following main assumptions (see also [18, 9]) will be employed.

Assumption 2. The linear operator A is the generator of a strongly continuous semigroup e^{tA} in X .

Assumption 3. Suppose that (4.1) possesses a sufficiently smooth solution $u : [0, T] \rightarrow X$, with derivatives in X and that the nonlinearity $g : X \rightarrow X$ is sufficiently often Fréchet differentiable in a strip along the exact solution. All occurring derivatives are supposed to be uniformly bounded.

By using a standard perturbation result in [21, Chap. 3.1], it is easy to infer from Assumptions 1 and 2 that the Jacobian

$$J = J(u) = \frac{\partial F}{\partial u}(u) = A + g'(u) \quad (4.2)$$

also generates a strongly continuous semigroup. This implies that there exist constants C and ω such that the bound

$$\|e^{tJ}\|_{X \leftarrow X} \leq Ce^{\omega t}, \quad t \geq 0 \quad (4.3)$$

holds uniformly in a neighborhood of the exact solution. As a consequence of the bound (4.3), one can see that the coefficients $\varphi_1(hJ_n)$, $\varphi_1(c_2hJ_n)$ and $b_2(hJ_n)$ of the 2-stage exponential Rosenbrock scheme (2.5) are bounded operators. Assumption 2 further implies that the Jacobian (4.2) and $g(u)$ are both locally Lipschitz in a strip along the exact solution u . In particular, in a neighborhood of the exact solution we have

$$\|J(u) - J(v)\|_{X \leftarrow X} = \|g'(u) - g'(v)\|_{X \leftarrow X} \leq L\|u - v\|. \quad (4.4)$$

4.1. Local error and relaxing stiff order conditions for 2-stage methods

As we are interested in constructing a superconvergent 2-stage exponential Rosenbrock scheme (2.5) for solving (4.1), one has to find the right coefficient $b_2(hJ_n)$ which satisfies the *stiff* order conditions for methods of order 4. In the following we will show that this can be done by using the new and simplified stiff order conditions for exponential Rosenbrock methods of order 4 given in [9, 19] and relaxing one of them. For convenience, we display the local error expansion of 2-stage methods, which can be obtained at once by using the result of the local error for s -stage methods given in [9, Sec.3.3], as follows

$$\begin{aligned} \tilde{e}_{n+1} = & h^3 \left(b_2(h\tilde{J}_n) \frac{c_2^2}{2!} - \varphi_3(h\tilde{J}_n) \right) \tilde{g}_n''(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n') \\ & + h^4 \left(b_2(h\tilde{J}_n) \frac{c_2^3}{3!} - \varphi_4(h\tilde{J}_n) \right) (\tilde{g}_n^{(3)}(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n', \tilde{u}_n') + 3\tilde{g}_n''(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n'')) + \mathcal{O}(h^5). \end{aligned} \quad (4.5)$$

Here

$$\tilde{J}_n = A + g'(\tilde{u}_n), \quad \tilde{g}_n(u) = F(u) - \tilde{J}_n u = g(u) - g'(\tilde{u}_n)u. \quad (4.6)$$

Note that since $\tilde{g}_n'(\tilde{u}_n) = 0$ and $\tilde{g}_n^{(k)}(u) = g^{(k)}(u)$ ($k \geq 2$), one can actually replace (4.5) by

$$\begin{aligned} \tilde{e}_{n+1} = & h^3 \left(b_2(h\tilde{J}_n) \frac{c_2^2}{2!} - \varphi_3(h\tilde{J}_n) \right) g''(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n') \\ & + h^4 \left(b_2(h\tilde{J}_n) \frac{c_2^3}{3!} - \varphi_4(h\tilde{J}_n) \right) (g^{(3)}(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n', \tilde{u}_n') + 3g''(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n'')) + \mathcal{O}(h^5). \end{aligned} \quad (4.7)$$

Requiring $\tilde{e}_{n+1} = \mathcal{O}(h^5)$ retrieves the stiff order conditions for methods of order 4 (see [9]), which are written for 2-stage methods as

$$b_2(Z) \frac{c_2^2}{2!} = \varphi_3(Z), \quad b_2(Z) \frac{c_2^3}{3!} = \varphi_4(Z)$$

with Z denotes an arbitrary square matrix. However, this is impossible due to the fact that matrix functions $\varphi_3(Z), \varphi_4(Z)$ are linearly independent. We thus follow the similar remedy as presented in [9, Sec.4.2] in order to relax the stiff order conditions. First, one realizes that there exist bounded operators $\hat{b}_2(h\tilde{J}_n)$ and $\hat{\varphi}_4(h\tilde{J}_n)$ such that

$$b_2(h\tilde{J}_n) \frac{c_2^3}{3!} - \varphi_4(h\tilde{J}_n) = (b_2(0) \frac{c_2^3}{3!} - \varphi_4(0)) + h \left(\hat{b}_2(h\tilde{J}_n) \frac{c_2^3}{3!} - \hat{\varphi}_4(h\tilde{J}_n) \right) \tilde{J}_n. \quad (4.8)$$

This is due to the recurrence relation (2.3) for $\varphi_k(z)$ and the fact that $b_2(h\tilde{J}_n)$ is chosen as linear combinations of $\varphi_k(h\tilde{J}_n)$. Inserting (4.8) into (4.7) gives

$$\begin{aligned}\tilde{e}_{n+1} = & h^3 \left(b_2(h\tilde{J}_n) \frac{c_2^2}{2!} - \varphi_3(h\tilde{J}_n) \right) g''(\tilde{u}_n)(\tilde{u}'_n, \tilde{u}'_n) \\ & + h^4 \left(b_2(0) \frac{c_2^3}{3!} - \varphi_4(0) \right) (g^{(3)}(\tilde{u}_n)(\tilde{u}'_n, \tilde{u}'_n, \tilde{u}'_n) + 3g''(\tilde{u}_n)(\tilde{u}'_n, \tilde{u}''_n)) \\ & + h^5 \left(\widehat{b}_2(h\tilde{J}_n) \frac{c_2^3}{3!} - \widehat{\varphi}_4(h\tilde{J}_n) \right) \tilde{J}_n (g^{(3)}(\tilde{u}_n)(\tilde{u}'_n, \tilde{u}'_n, \tilde{u}'_n) + 3g''(\tilde{u}_n)(\tilde{u}'_n, \tilde{u}''_n)) + \mathcal{O}(h^5).\end{aligned}\tag{4.9}$$

This local error expansion brings us to the following result concerning the relaxing stiff order conditions for 2-stage methods of order 4.

Lemma 4.1. *Under Assumptions 2 and 3 and further assume that the operator A and the nonlinearity $g(u)$ in (4.1) are such that*

$$A(g^{(3)}(\tilde{u}_n)(\tilde{u}'_n, \tilde{u}'_n, \tilde{u}'_n) + 3g''(\tilde{u}_n)(\tilde{u}'_n, \tilde{u}''_n)) \tag{4.10}$$

is uniformly bounded on X , a 2-stage explicit exponential Rosenbrock method (2.5) has order of consistency five, i.e. $\tilde{e}_{n+1} = \mathcal{O}(h^5)$ if the following order conditions are fulfilled

$$b_2(Z)c_2^2 = 2\varphi_3(Z), \tag{4.11a}$$

$$b_2(0)c_2^3 = 6\varphi_4(0) \tag{4.11b}$$

with Z denotes an arbitrary square matrix. Moreover, the remainder term of \tilde{e}_{n+1} , which is hidden behind the Landau notation $\mathcal{O}(\cdot)$, is bounded by Ch^5 with a constant C that only depends on values that are uniformly bounded by the assumptions made, i.e., is independent of n and h .

Proof. It follows at once from Assumption 3 and the additional regularity assumption (4.10) that $\tilde{J}_n(g^{(3)}(\tilde{u}_n)(\tilde{u}'_n, \tilde{u}'_n, \tilde{u}'_n) + 3g''(\tilde{u}_n)(\tilde{u}'_n, \tilde{u}''_n))$ is also uniformly bounded on X (since $\tilde{J}_n = A + g'(\tilde{u}_n)$). In view of (4.9), the conclusion of Lemma 4.1 is thus verified by using the Assumptions 2–3 and the given order conditions in (4.11). \square

Remark 2. The additional smoothness condition (4.10) is often fulfilled for many semilinear parabolic PDEs such as reaction-diffusion equations, the Allen-Cahn equation and the Chafee-Infante problem [22, Chap. 5], where the operator A is the strongly second-order elliptic differential operator (e.g. the Laplacian or the gradient). In particular, for such problems, one can show that Assumption 3 implies (4.10). For more details, we refer to [9, Example 4.1].

4.2. Derivation of a fourth-order 2-stage stiffly accurate scheme

Solving the order conditions in (4.11) gives $b_2(0) = \frac{2\varphi_3(0)}{c_2^2} = \frac{6\varphi_4(0)}{c_2^3}$ which implies $c_2 = \frac{3\varphi_4(0)}{\varphi_3(0)} = \frac{3}{4}$ and thus $b_2(Z) = \frac{32}{9}\varphi_3(Z)$. This is the unique solution of (4.11). Inserting this result into (2.5) we obtain the following 2-stage scheme which will be called **exprb42**:

$$U_{n2} = u_n + \frac{3}{4}h\varphi_1\left(\frac{3}{4}hJ_n\right)F(u_n), \quad (4.12a)$$

$$u_{n+1} = u_n + h\varphi_1(hJ_n)F(u_n) + h\frac{32}{9}\varphi_3(hJ_n)(g_n(U_{n2}) - g_n(u_n)). \quad (4.12b)$$

The convergence of this scheme will be stated in the next section.

4.3. Stability and convergence result

It is shown in [18, Sec. 3.3] that the following stability bound

$$\left\| \prod_{j=0}^{n-k} e^{hJ_{n-j}} \right\|_{X \leftarrow X} \leq C_S, \quad t_0 \leq t_k \leq t_n \leq T \quad (4.13)$$

is the key to show the convergence of exponential Rosenbrock methods (2.1). The good thing here is that the constant C_S in (4.13) is uniform in k and n despite the fact that J_n varies from step to step.

With the help of (4.13), one can prove that **exprb42** converges with global order 4 by using the same techniques presented in the recent work [9, Sec. 4] (presenting the convergence results for methods of orders up to 5). For convenience for the reader, below we recall some of the important results which can be applied directly to our case ($s = 2$, $c_2 = \frac{3}{4}$, $b_2(hJ_n) = \frac{32}{9}\varphi_3(hJ_n)$). However, we will omit other details of their proofs.

Let $e_{n+1} = u_{n+1} - u(t_{n+1}) = u_{n+1} - \tilde{u}_{n+1}$ denote the global error of the scheme (4.12). One can show that it satisfies

$$e_{n+1} = e^{hJ_n}e_n + hP_n + \tilde{e}_{n+1}, \quad e_0 = 0 \quad (4.14)$$

with

$$\begin{aligned} P_n = & \varphi_1(hJ_n)(g_n(u_n) - g_n(\tilde{u}_n)) + (\varphi_1(hJ_n) - \varphi_1(h\tilde{J}_n))F(\tilde{u}_n) \\ & + b_2(hJ_n)(g_n(U_{n2}) - g_n(u_n)) - b_2(h\tilde{J}_n)(\tilde{g}_n(\bar{U}_{n2}) - \tilde{g}_n(\tilde{u}_n)). \end{aligned} \quad (4.15)$$

Under Assumptions 2 and 3, the following estimate

$$\|P_n\| \leq C\|e_n\| + C\|e_n\|^2 + Ch^6 \quad (4.16)$$

holds true as a direct result of Lemma 4.5 in [9].

We are now at the final stage of formulating our convergence result.

Theorem 4.1. *Let the initial value problem (4.1) satisfy the Assumptions of Lemma 4.1. Then, the numerical solution u_n of the 2-stage explicit exponential Rosenbrock method **exprb42** satisfies the error bound*

$$\|u_n - u(t_n)\| \leq Ch^4 \quad (4.17)$$

uniformly on $t_0 \leq t_n = t_0 + nh \leq T$ with a constant C that depends on $T - t_0$, but is independent of n and h .

Proof. Solving the recursion (4.14) gives

$$e_n = h \sum_{k=0}^{n-1} \prod_{j=1}^{n-k-1} e^{hJ_{n-j}} \left(P_k + \frac{1}{h} \tilde{e}_{k+1} \right). \quad (4.18)$$

The result of Lemma 4.1 shows that the local error of **exprb42** satisfies $\tilde{e}_{k+1} = \mathcal{O}(h^5)$. Next, we use the stability estimate (4.13) and the bound (4.16) to get

$$\|e_n\| \leq C \sum_{k=0}^{n-1} h (\|e_k\| + \|e_k\|^2 + h^4). \quad (4.19)$$

The desired bound (4.17) follows by an application of a discrete Gronwall lemma (see [23]) to (4.19). \square

5. Adaptive time-stepping schemes

It should be mentioned that the newly constructed schemes **exprb42** (see (4.12)) and **exprb42N** (see (3.16)) can also be implemented with variable stepsizes. Indeed one can use the standard way as employed in [18, 9] (for other exponential Rosenbrock schemes) that is to consider (2.5) together with an embedded scheme of lower order

$$\hat{u}_{n+1} = u_n + h\varphi_1(hJ_n)F(u_n) + h\hat{b}_2(hJ_n)(g_n(U_{n2}) - g_n(u_n)) \quad (5.1)$$

which uses the same internal stage U_{n2} . It is clear that a 2-stage method of order 3 requires $b_2(Z) = \frac{2\varphi_3(Z)}{c_2^2}$ for any node $c_2 \neq 0$. For $c_2 = 3/4$ that is uniquely determined (so is U_{n2}) by the construction of the two new schemes, it has been shown that such a 2-stage method can even attain order 4. This implies that it is impossible to embed **exprb42** with a 2-stage method of order 3. The fact that **exprb42N** has the same U_{n2} as **exprb42**, we thus consider to embed both of them with a second-order error estimate, which is the exponential Rosenbrock-Euler method (so $\hat{b}_2(hJ_n) = 0$). For later use in our numerical experiments, we display **exprb42** and **exprb42N** (for variable stepsizes implementation) in reduced Butcher tableau (see [13, Sect.2]) as follows

exprb42N:

$$\begin{array}{c|c} \frac{3}{4} & \\ \hline & \begin{array}{cc} 8 & 48 \\ -\frac{8}{27}\varphi_1 + \frac{48}{27}\varphi_2 & \\ 0 & \end{array} \end{array}$$

exprb42:

$$\begin{array}{c|c} \frac{3}{4} & \\ \hline & \begin{array}{c} 32 \\ \frac{32}{9}\varphi_3 \\ 0 \end{array} \end{array}.$$

6. Numerical experiments

In this section we verify our convergence results and demonstrate the efficiency of the new integrators **exprb42N** and **exprb42**. To this aim, we carry out numerical experiments on a set of test problems (see below). First, we discuss the implementation of the new integrators.

6.1. Implementation and test problems

6.1.1. Implementation

The implementation of exponential integrators (in particular, the new integrators **exprb42N** and **exprb42**) requires computing the action of matrix functions $\varphi_k(hJ_n)$ on vectors v_k . With the recent developments of numerical linear algebra in computing matrix functions (see, for example [24, 25, 26]), this can be done efficiently. In order to take advantages of computing a linear combination of terms like $\sum_{k=1}^p \varphi_k(hJ_n)v_k$ (by one single evaluation) and computational time, we use here the adaptive Krylov technique proposed in [25, 27]. For variable step sizes implementation, the error estimate $\mathbf{err} = u_{n+1} - \hat{u}_{n+1}$ (see [2, Chapter IV.8]) will be used to control time steps. All the simulations are run in MATLAB.

Next, we give a list of test problems including both nonstiff and stiff differential equations that fit in the framework.

6.1.2. Nonstiff problems

Example 6.1. Consider an example from Astronomy-the restricted three body problem (see [1, 28]):

$$y_1'' = -\frac{y_1}{(y_1^2 + y_2^2)^{3/2}}, \quad y_2'' = -\frac{y_2}{(y_1^2 + y_2^2)^{3/2}}, \quad t \in [0, 10]. \quad (6.1)$$

The equation of motion above can be written as a system of first-order differential equations as $y_1' = y_3$, $y_2' = y_4$, $y_3' = -y_1/r^3$, $y_4' = -y_2/r^3$ with $r = \sqrt{y_1^2 + y_2^2}$. For this simple case (which can be considered as a two-body orbit problem) the exact solution is known, that is $y(t) = [\cos(t), \sin(t), -\sin(t), \cos(t)]$.

Example 6.2. Consider the van der Pol equation (see [1]):

$$\begin{aligned} y_1' &= y_2, \\ y_2' &= (1 - y_1^2)y_2 - y_1, \quad t \in [0, 2] \\ y_1(0) &= 2, \quad y_2(0) = 0. \end{aligned} \quad (6.2)$$

Since the exact solution of (6.2) is unknown, we compute its reference solution by using a nonstiff solver such as *ode45* with $\text{ATOL}=\text{RTOL}=10^{-14}$.

6.1.3. Stiff problems

Example 6.3. Consider the one-dimensional semilinear parabolic problem (see [6])

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = \frac{1}{1 + u^2} + \Phi(x, t) \quad (6.3)$$

for $u = u(x, t)$ on the unit interval $[0, 1]$ and $t \in [0, 1]$, subject to homogeneous Dirichlet boundary conditions. The source function Φ is chosen in such a way that the exact solution of the problem is $u(x, t) = x(1 - x)e^t$.

In order to solve (6.3) numerically, the first step is to discretize it in space by standard finite differences with $M = 199$ (inner) grid points. This yields a very stiff system of the form (4.1) (with $\|A\|_\infty = 1.5999e + 05$). Then we use our new integrators to integrate this ODE system in time with constant step sizes. Let $U_i^{n,h} \approx u(x_i, t_n)$ denote the numerical solution at $t_n = nh$ and grid point $x_i = \frac{i}{200}$ and let \bar{u}_i denote a reference solution of the spatially discrete

problem at time $t = 1$ and grid point x_i , computed with sufficiently small time steps. Note that for this example, since we know the exact solution, one can take $\bar{u}_i = x_i(1 - x_i)$. The time integration errors $U^{N,1/N} - \bar{u}$ are measured in the maximum norm $\max_{1 \leq i \leq 199} |U_i^{N,1/N} - \bar{u}_i|$.

Example 6.4. Consider the two-dimensional advection-diffusion-reaction equation (see, for example, [18, 9])

$$\frac{\partial u}{\partial t} = 0.01\Delta u + 10\nabla u + 100u\left(u - \frac{1}{2}\right)(1 - u) \quad (6.4)$$

for $u = u(x, y, t)$ on the unit square $\Omega = [0, 1]^2$ with the initial value

$$u(x, y, 0) = 0.3 + 256(x(1 - x)y(1 - y))^2,$$

subject to homogeneous Neumann boundary conditions. Here Δ and ∇ denote the Laplacian and the gradient vector field in two dimensions, respectively. Discretizing (6.4) in space by standard finite differences using 101 grid points in each direction with meshwidth $dx = dy = 1/100$ yields a mildly stiff system of the form (4.1) (with $\|A\|_\infty = 2.4e + 03$). For the time integration of this resulting system of ODEs, we use our new integrators. Since the exact solution of (6.4) is unknown, a reliable reference solution is computed by using sufficiently small time steps (one can also use the stiff solver `ode15s` with $ATOL=RTOL=10^{-14}$). As done for Example 6.3, the time integration errors are measured in a discrete maximum norm at the final time $T = 0.08$.

6.2. Accuracy verification and performance comparison

The purpose of giving the two nonstiff problems in Examples 6.1 whose exact solution is known and in Examples 6.2, whose exact solution is unknown, is just to verify the order 4 of the integrator `exprb42N` (satisfying the classical order conditions). However, we also display the order plots of the stiff integrator `exprb42` in Fig. 1 (in a double-logarithmic diagram). In this experiment, we use constant step sizes that correspond to the number of time steps that are $N = 64, 128, 256, 512$. The diagrams clearly shows a perfect agreement with Theorem 3.1. It is observed that with the same number of time steps `exprb42` even gets a bit more accuracy than `exprb42N` for Example 6.1. For Example 6.2 both integrators give almost identical results.

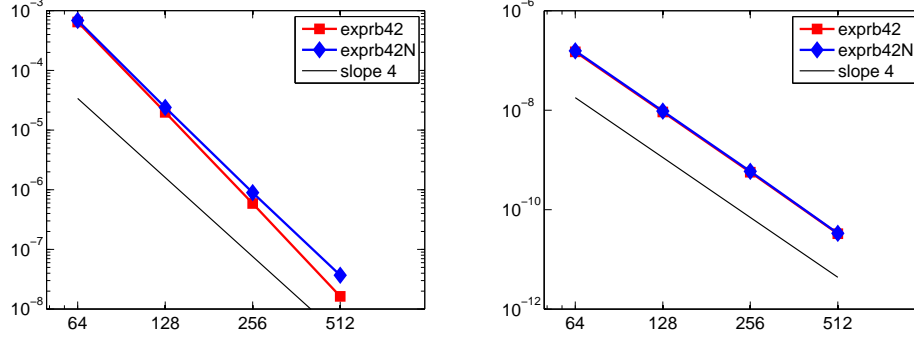


Figure 1: Order plots of **exprb42N** and **exprb42** when applied to Example 6.1 (left) and Example 6.2 (right). The errors at time $t = 10$ (left) and $t = 2$ (right) are plotted as functions of the number of time steps N . For comparison, a straight line with slope 4 is added.

For the two stiff problems in Examples 6.3 and 6.4, we also check the sharpness of the error bound given in Theorem 4.1. Since the problems are stiff, it is interesting to compare the two new 2-stage fourth-order explicit integrators with the 2-stage Gauss–Legendre scheme—the only existing class of 2-stage fourth-order method (see [2]). We will call it as **Gauss42** for the rest of the paper. Since **Gauss42** is an implicit Runge–Kutta method, we use the simplified Newton iterations as suggested in [2] for its implementation.

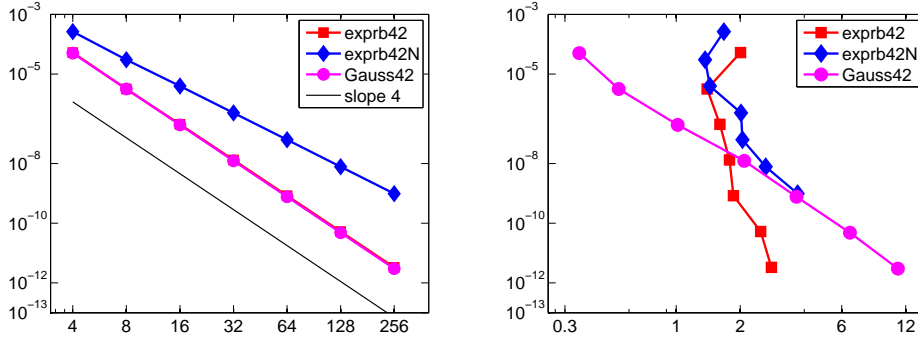


Figure 2: Order plots (left) and total CPU times (right) of **exprb42N**, **exprb42**, and **Gauss42** when applied to Example 6.3. The errors at time $t = 1$ are plotted as functions of the number of time steps $N = 4, 8, 16, 32, 64, 128, 256$ (left) and the total CPU time in second (right). For comparison, a straight line with slope 4 is added.

As seen from the left diagram in Fig. 2, while **exprb42N** suffers from order reduction when applied to the very stiff problem in Example 6.3, the two stiff solvers **exprb42** and **Gauss42** achieve perfectly order 4 and give almost

identical global errors for a given number of time steps. In the right diagram we plot the total CPU time versus global error. It turns out that **exprb42** is the fastest one for more stringent global error tolerances (much faster than **Gauss42**). Moreover, as the number of time steps increases, the CPU time of **exprb42** does not increase much while **Gauss42** increases CPU time with rate in an approximately linear manner. This can be explained as **Gauss42** requires solving a nonlinear system of equations at every step.

In Fig. 3, we again use constant step sizes (corresponding to the number of time steps $N = 32, 64, 128, 256, 512$) to verify the achieved orders of the two new integrators when applied to Example 6.4. Along with **Gauss42** we also added the best fourth-order exponential Rosenbrock scheme, **pexprb43**, which requires 3 stages (see [13]) to this comparison. The left precision diagram clearly confirms that all integrators are indeed of order 4, meaning that **exprb42N** does not suffer from order reduction for this mildly stiff problem. In addition, we see that the three fourth-order exponential integrators even offer more accuracy than **Gauss42** for a given number of time steps. For this two-dimensional problem, the right precision diagram indicates a huge computational saving of the three exponential schemes over **Gauss42**. This is again due to the implicitness of scheme **Gauss42**. Furthermore, it is observed that both **exprb42N** and **exprb42** are a bit faster than **pexprb43**.

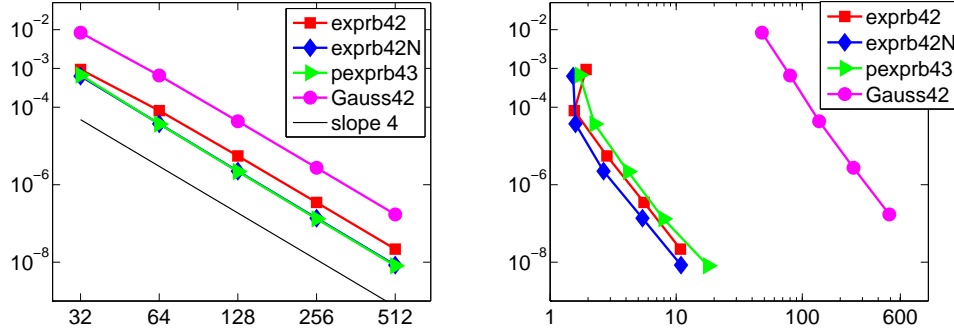


Figure 3: Order plots (left) and total CPU times (right) of **exprb42N**, **exprb42**, **pexprb43**, and **Gauss42** when applied to Example 6.4. The errors at time $t = 0.08$ are plotted as functions of the number of time steps $N = 32, 64, 128, 256, 512$ (left) and the total CPU time in second (right). For comparison, a straight line with slope 4 is added.

Next, we implement the new integrators using variable step sizes codes. In Fig. 4, using the same tolerances $\text{ATOL} = \text{RTOL} = 10^{-4}, 10^{-4.5}, \dots, 10^{-6}$ we plot the achieved accuracy as a function of the required number of time

steps. The results are compared again with those of `pexprb43`.

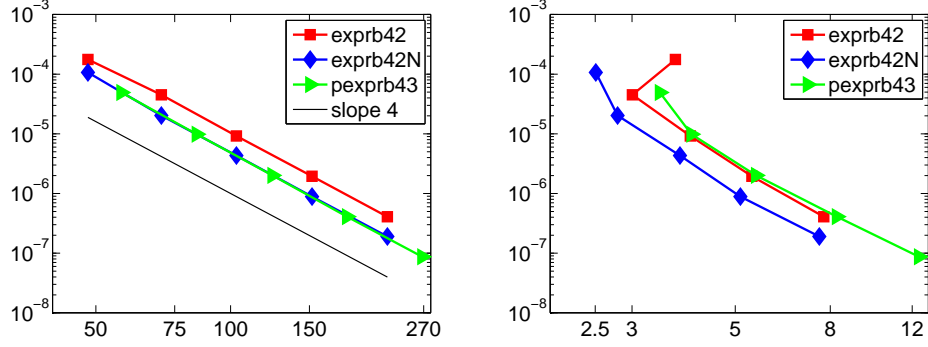


Figure 4: Number of time steps versus accuracy (left) and the total CPU time versus accuracy (right) for the advection-diffusion-reaction Example 6.4 for $t = 0.08$. The errors are measured in a discrete maximum norm.

The precision diagrams in Fig. 4 indicates that `pexprb43` gets a bit more accuracy but takes more number of time steps as well as requires more CPU time than `exprb42N`, `exprb42`. This observation is fairly comparable with the experiments using constant step sizes in Fig. 3.

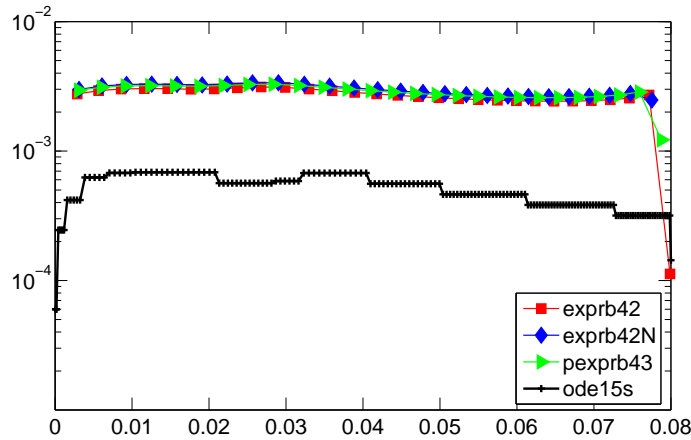


Figure 5: Time versus step sizes for the advection-diffusion-reaction Example 6.4 for an accuracy of about 0.001 at $t = 0.08$.

Finally, we fix a final accuracy of about 0.001 at $t = 0.08$ (by choosing appropriate tolerances) for integrators `exprb42N`, `exprb42`, `pexprb43` as well as a well-established and widely-use code-the stiff solver `ode15s` in order to

compare their chosen step sizes. As seen from Fig. 5, the new integrators `exprb42N` and `exprb42` use about the same steps as `pexprb43` (28-30 steps) and take much larger time steps compared to `ode15s` (165 steps). Overall, we conclude that both `exprb42N` and `exprb42` perform quite well and they certainly beat the implicit methods such as `Gauss42` and `ode15s` for the advection-diffusion-reaction Example 6.4.

References

- [1] E. Hairer, S. Nørsett, G. Wanner, Solving Ordinary Differential Equations I: Nonstiff Problems, Springer, 1993.
- [2] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, Springer, New York, 1996.
- [3] M. Hochbruck, A. Ostermann, Exponential integrators, *Acta Numerica* 19 (2010) 209–286.
- [4] M. Hochbruck, C. Lubich, H. Selhofer, Exponential integrators for large systems of differential equations, *SIAM J. Sci. Comput.* 19 (1998) 1552–1574.
- [5] S. Cox, P. Matthews, Exponential time differencing for stiff systems, *J. Comput. Phys.* 176 (2002) 430–455.
- [6] M. Hochbruck, A. Ostermann, Explicit exponential Runge–Kutta methods for semilinear parabolic problems, *SIAM J. Numer. Anal.* 43 (2005) 1069–1090.
- [7] M. Tokman, Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods, *J. Comput. Phys.* 213 (2006) 748–776.
- [8] V. T. Luan, A. Ostermann, Stiff order conditions for exponential Runge–Kutta methods of order five, in: H. B. et al. (Ed.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, Springer, 2014, pp. 133–143.
- [9] V. T. Luan, A. Ostermann, Exponential Rosenbrock methods of order five—construction, analysis and numerical comparisons, *J. Comput. Appl. Math.* 255 (2014) 417–431.

- [10] V. T. Luan, A. Ostermann, Explicit exponential Runge–Kutta methods of high order for parabolic problems, *J. Comput. Appl. Math.* 256 (2014) 168–179.
- [11] P. Tranquilli, A. Sandu, Exponential-krylov methods for ordinary differential equations, *J. Comput. Phys.* 278 (2014) 31–46.
- [12] G. Rainwater, M. Tokman, A new class of split exponential propagation iterative methods of Runge–Kutta type (sEPIRK) for semilinear systems of ODEs, *J. Comput. Phys.* 269 (2014) 40–60.
- [13] V. T. Luan, A. Ostermann, Parallel exponential Rosenbrock methods, *Comput. Math. Appl.* 71 (2016) 1137–1150.
- [14] V. T. Luan, M. Tokman, G. Rainwater, Preconditioned implicit-exponential (IMEXP) time integrators for stiff differential equations, *arXiv:1605.02858*.
- [15] V. T. Luan, High-order exponential integrators, Ph.D. thesis, University of Innsbruck (2014).
- [16] M. Hochbruck, A. Ostermann, Explicit integrators of Rosenbrock-type, *Oberwolfach Reports* 3 3 (2006) 1107–1110.
- [17] D. A. Pope, An exponential method of numerical integration of ordinary differential equations, *Comm. ACM* 6 (1963) 491–493.
- [18] M. Hochbruck, A. Ostermann, J. Schweitzer, Exponential rosenbrock-type methods, *SIAM J. Numer. Anal.* 47 (2009) 786–803.
- [19] V. T. Luan, A. Ostermann, Exponential B-series: The stiff case, *SIAM J. Numer. Anal.* 51 (2013) 3431–3445.
- [20] K. Engel, R. Nagel, *One-parameter Semigroups for Linear Evolution Equations*, Springer, New York, 2000.
- [21] A. Pazy, *Semigroups of Linear Operators and Applications to Partial Differential Equations*, Springer, New York, 1983.
- [22] D. Henry, *Geometric Theory of Semilinear Parabolic Equations*, Vol. 840 of *Lecture Notes in Mathematics*, Springer-Verlag Berlin Heidelberg, 1981.

- [23] E. Emmrich, Stability and error of the variable two-step BDF for semi-linear parabolic problems, *J. Appl. Math. Comput.* 19 (1) (2005) 33–55.
- [24] A. H. Al-Mohy, N. J. Higham, Computing the action of the matrix exponential with an application to exponential integrators, *SIAM J. Sci. Comput.* 33 (2011) 488–511.
- [25] J. Niesen, W. M. Wright, Algorithm 919: A Krylov subspace algorithm for evaluating the φ -functions appearing in exponential integrators, *ACM Trans. Math. Software.* 38, Article 22 (2012).
- [26] M. Caliari, P. Kandolf, A. Ostermann, S. Rainer, Comparison of methods for computing the action of the matrix exponential, *BIT Numer. Math.* 52 (1) (2014) 113–128.
- [27] M. Tokman, J. Loffeld, P. Tranquilli, New adaptive exponential propagation iterative methods of Runge–Kutta type, *SIAM J. Sci. Comput.* 34 (2012) A2650–A2669.
- [28] H. Podhaisky, R. Weiner, J. Wensch, High order explicit two-step Runge–Kutta methods for parallel computers, *J. Comput. Info. Tech. (CIT)* 1 (2000) 13–18.